

# Datos

---

- Tarea 1
- Laboratorio de Lenguajes Formales de Programación
- Linda Madelin Fabiola Quelex Sep
- 201403745

# Código fuente

---

```
public class Tarea2 {  
    private final int REPETICIONES =5;  
    public static void main(String []args){  
        int i=0;  
        while(i<REPETICIONES){  
            System.out.println("Repetición NO. " + (i+1));  
            i++;  
        }  
    }  
}
```

# Lexemas

---

- public
- class
- Tarea2
- {
- private
- final
- int
- REPETICIONES
- =
- 5
- ;
- public
- static
- void
- main
- (
- String
- [
- ]
- args
- )
- {

- int
- i
- =
- 0
- ;
- while
- (
- i
- <
- REPETICIONES
- )
- {
- System
- .
- out
- .
- println
- (
- " Repetición NO. "
- +
- (
- i
- +
- 1
- )
- )
- ;
- i
- +
- +
- ;
- }
- }
- }

# Definición de tokens

Token	Descripción	Patrón
reservada_public	Palabra reservada	public
reservada_class	Palabra reservada	class
identificador	Cualquier identificador del lenguaje	(\\[a-zA-Z])(\\[a-zA-Z0-9])
llave_abierta	Llave abierta	{

Token	Descripción	Patrón
llave_cerrada	Llave cerrada	}
reservada_private	Palabra reservada	private
reservada_final	Palabra reservada	final
tipo_int	Tipo de dato entero	int
operador_igual	Operador de asignación	=
Dato_int	Dato tipo entero	^\d+\$
punto_coma	Punto y coma	;
reservada_static	Palabra reservada	static
reservada_void	Palabra reservada	void
parentesis_abierto	Paréntesis abierto	(
parentesis_cerrado	Paréntesis cerrado	)
tipo_string	Tipo de dato String	String
corchete_open	Corchete abierto	[
corchete_close	Corchete cerrado	]
iterativo	Ciclo while	While
menor_que	Operador menor que	<
punto	Operador	.
suma	Operador suma	+
dato_String	Dato tipo String	"([^\"] \\( \"))*)"

## Análisis léxico tarea

Lexema	Token
public	reservada_public
class	reservada_class
Tarea2	identificador
{	llave_abierta
private	reservada_private
final	reservada_final
int	tipo_int
REPETICIONES	identificador
=	operador_igual

Lexema	Token
5	Dato_int
;	punto_coma
public	reservada_public
static	reservada_static
void	reservada_void
main	identificador
(	parentesis_abierto
String	tipo_string
[	corchete_open
]	corchete_close
args	identificador
)	parentesis_cerrado
{	llave_abierta
int	tipo_int
i	identificador
=	operador_igual
0	Dato_int
;	punto_coma
while	iterativo
(	parentesis_abierto
i	identificador
<	menor_que
REPETICIONES	identificador
)	parentesis_cerrado
{	llave_abierta
System	identificador
.	punto
out	identificador
.	punto
println	identificador
(	llave_abierta
" Repetición NO. "	dato_String

Lexema	Token
+	suma
(	llave_abierta
i	identificador
+	suma
1	Dato_int
)	parentesis_cerrado
)	parentesis_cerrado
;	punto_coma
i	identificador
+	suma
+	suma
;	punto_coma
}	llave_cerrada
}	llave_cerrada
}	llave_cerrada